Fachhochschule Wedel
Prof. Dr. Uwe Schmidt

7. August 1996

Aufgaben zur Klausur **Grundlagen der Programmierung**, **Software Engineering** und **Logische Programmierung** im SS 96 (WI03)

Zeit: 120 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 9 Seiten

Aufgabe 1:

Beweisen Sie durch Transformation, daß die folgende Formel ein Satz der Aussagenlogik ist. Begründen Sie die einzelnen Beweisschritte.

$$\neg(c \Rightarrow b) \Rightarrow \neg(a \Rightarrow b) \vee (a \oplus c)$$

Nutzen Sie diese Seite für die Kladde, die nächste Seite für die fertige Lösung.

$\neg(c \Rightarrow b) \Rightarrow \neg(a \Rightarrow b) \vee (a \oplus c)$

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

$\Leftrightarrow$ *Begründung :* ....

....

Aufgabe 2:

Gegeben seien die Felder f und g

var

f : array $[0..n-1]$ of Z
g : array $[0..n-1]$ of Z

mit je n Elementen $(n > 0)$ und die folgenden prädikatenlogischen Formeln

1. $\forall\, 0 \leq i < n \bullet f[i] \geq 0 \Leftrightarrow g[i] \geq 0$
2. $\forall\, 0 \leq i < n \bullet f[i] < 0 \Leftrightarrow g[i] < 0$
3. $\forall\, 0 \leq i < n \bullet f[i] \geq 0 \Leftrightarrow g[i] < 0$
4. $\forall\, 0 \leq i < n \bullet f[i] \geq 0 \oplus g[i] \geq 0$
5. $\forall\, 0 \leq i < n \bullet f[i] < 0 \Leftrightarrow g[i] > 0$
6. $\forall\, 0 \leq i < n \bullet f[i] \geq 0 \Rightarrow g[i] \geq 0$
7. $\forall\, 0 \leq i < n \bullet f[i] \geq 0 \wedge g[i] \geq 0$
8. $\forall\, 0 \leq i < n \bullet f[i] < 0 \vee g[i] \geq 0$
9. $\forall\, 0 \leq i < n \bullet f[i] \leq 0 \vee g[i] > 0$
10. $\forall\, 0 \leq i, j < n \bullet i = j \Rightarrow f[i] > f[j]$
11. $\forall\, 0 \leq i, j < n \bullet i = j \Rightarrow f[i] > g[j]$
12. $\forall\, 0 \leq i, j < n \bullet i \neq j \wedge f[i] \leq g[j]$
13. $\forall\, 0 \leq i, j < n \bullet i \neq j \vee f[i] > g[j]$
14. $\forall\, 0 \leq i, j < n \bullet f[i] - g[j] > 0 \vee i \neq j$

Geben sie für die folgenden Aussagen die Nummer(n) von **gleichwertigen** Formeln an, Mehrfachnennungen sind möglich, gibt es keine Formel tragen Sie 0 an die vorgesehene Stelle ein.

1. An jeder Position in f steht ein echt größerer Wert als an der gleichen Position in g.

   ....

2. Wenn in f an einer Position ein positiver Wert gespeichert ist, dann auch in g an der gleichen Position.

   ....

3. Alle in f und g an gleicher Position gespeicherten Elemente haben das gleiche Vorzeichen.

   ....

4. Alle in f und g an gleicher Position gespeicherten Elemente haben verschiedene Vorzeichen.

   ....

5. `false`

   ....

6. `true`

   ....

Aufgabe 3:

Gegeben sei die folgende rekursive Funktion

$f(x : \mathsf{N}_1) : \mathsf{Z}$
  if $x = 1$
  then 0
  else
    if x mod $2 = 0$
    then $f(x$ div $2) - 1$
    else $f(3 * x + 1) + 1$

Transformieren Sie diese Funktion auf systematische Art und Weise in eine gleichwertige Funktion, die mit einer while Schleife arbeitet.

Die Transformation erfordert einen Zwischenschritt, führen Sie diesen in Kladde durch.

Das Ergebnis:

....

....

....

....

....

....

....

....

....

....

Aufgabe 4:

Gegeben sei folgendes Programm zur Berechnung von Wegen in einem Graphen, die Kanten in einem Graphen werden durch kante/3 mit Anfangs- und Endknoten als 1. und 2. Argument repräsentiert, die Länge einer Kante von dem Anfangsknoten zum Endknoten ist im 3. Argument gespeichert.

Das Prädikat weg(+AnfangsKnoten, +EndKnoten) testet, ob es einen Weg von einem Anfangsknoten zu einem Endknoten gibt.

```
kante(a, b, 5).
kante(a, c, 3).
kante(b, c, 2).
kante(b, a, 1).
kante(c, d, 4).

weg(X, Y) :-
        kante(X, Y, _).

weg(X, Y) :-
        kante(X, Z, _),
        weg(Z, Y).
```

Erweitern Sie das Prädikat weg/2 zu einem Prädikat weg/3, das die Länge des gefundenen Weges berechnent:

weg(+AnfangsKnoten, +EndKnoten, -Laenge):

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

Erweitern Sie das Prädikat weg/2 auf eine 2. Art so, daß es testet, ob es einen Weg mit einer Länge von höchstens Maxlen in dem Graphen gibt:

weg1(+AnfangsKnoten, +EndKnoten, +Maxlen):

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

1. Wieviele Lösungen bekommt man für die Anfrage `?- weg(a,d,L)`

   ....

   und wie sieht die 1. Lösung aus

   ....

2. Wieviele Lösungen bekommt man für die Anfrage `?- weg1(a,d,8)`

   ....

   und wie sieht die 1. Lösung aus

   ....

FACHHOCHSCHULE WEDEL
Prof. Dr. Uwe Schmidt

7. August 1996

Aufgaben zur Klausur **Grundlagen der Programmierung** und **Software Engineering** im SS 96 (II13)

Zeit: 120 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 7 Seiten

---

Aufgabe 1:

Beweisen Sie durch Transformation, daß die folgende Formel ein Satz der Aussagenlogik ist. Begründen Sie die einzelnen Beweisschritte.

$$\neg(c \Rightarrow b) \Rightarrow \neg(a \Rightarrow b) \vee (a \oplus c)$$

Nutzen Sie diese Seite für die Kladde, die nächste Seite für die fertige Lösung.

$\neg(c \Rightarrow b) \Rightarrow \neg(a \Rightarrow b) \vee (a \oplus c)$

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

$\Leftrightarrow$ *Begründung :* . . . .

. . . .

Aufgabe 2:

Gegeben seien die Felder f und g

var

f : array $[0..n-1]$ of Z
g : array $[0..n-1]$ of Z

mit je n Elementen ($n > 0$) und die folgenden prädikatenlogischen Formeln

1. $\forall\, 0 \le i < n \bullet f[i] \ge 0 \Leftrightarrow g[i] \ge 0$
2. $\forall\, 0 \le i < n \bullet f[i] < 0 \Leftrightarrow g[i] < 0$
3. $\forall\, 0 \le i < n \bullet f[i] \ge 0 \Leftrightarrow g[i] < 0$
4. $\forall\, 0 \le i < n \bullet f[i] \ge 0 \oplus g[i] \ge 0$
5. $\forall\, 0 \le i < n \bullet f[i] < 0 \Leftrightarrow g[i] > 0$
6. $\forall\, 0 \le i < n \bullet f[i] \ge 0 \Rightarrow g[i] \ge 0$
7. $\forall\, 0 \le i < n \bullet f[i] \ge 0 \wedge g[i] \ge 0$
8. $\forall\, 0 \le i < n \bullet f[i] < 0 \vee g[i] \ge 0$
9. $\forall\, 0 \le i < n \bullet f[i] \le 0 \vee g[i] > 0$
10. $\forall\, 0 \le i, j < n \bullet i = j \Rightarrow f[i] > f[j]$
11. $\forall\, 0 \le i, j < n \bullet i = j \Rightarrow f[i] > g[j]$
12. $\forall\, 0 \le i, j < n \bullet i \ne j \wedge f[i] \le g[j]$
13. $\forall\, 0 \le i, j < n \bullet i \ne j \vee f[i] > g[j]$
14. $\forall\, 0 \le i, j < n \bullet f[i] - g[j] > 0 \vee i \ne j$

Geben sie für die folgenden Aussagen die Nummer(n) von **gleichwertigen** Formeln an, Mehrfachnennungen sind möglich, gibt es keine Formel tragen Sie 0 an die vorgesehene Stelle ein.

1. An jeder Position in f steht ein echt größerer Wert als an der gleichen Position in g.

   ....

2. Wenn in f an einer Position ein positiver Wert gespeichert ist, dann auch in g an der gleichen Position.

   ....

3. Alle in f und g an gleicher Position gespeicherten Elemente haben das gleiche Vorzeichen.

   ....

4. Alle in f und g an gleicher Position gespeicherten Elemente haben verschiedene Vorzeichen.

   ....

5. `false`

   ....

6. `true`

   ....

Aufgabe 3:

Gegeben sei die folgende rekursive Funktion

$f(x : \mathsf{N}_1) : \mathsf{Z}$
  if $x = 1$
  then 0
  else
    if x mod $2 = 0$
    then $f(x \text{ div } 2) - 1$
    else $f(3 * x + 1) + 1$

Transformieren Sie diese Funktion auf systematische Art und Weise in eine gleichwertige Funktion, die mit einer while Schleife arbeitet.

Die Transformation erfordert einen Zwischenschritt, führen Sie diesen in Kladde durch.

Das Ergebnis:

....

....

....

....

....

....

....

....

....

....

Aufgabe 4:

Gegeben seien die folgenden Variablen:

var f : array $[0..n-1]$ of Z;
var b : B;
var i : N_0

Konstruieren Sie ein Programmstück, das folgendes Prädikat in der Variablen b berechnet.

$$\forall\, 0 < j < n \bullet f[j] \geq 0 \Leftrightarrow f[j-1] < 0$$

Das Programmstück soll systematisch mit den Techniken aus der Vorlesung aus dem $\forall$ Quantor abgeleitet werden, es soll so arbeiten, daß keine überflüssigen Berechnungen gemacht werden, nachdem das Resultat feststeht.

....

....

....

....

....

....

....

....